



International Journal for Innovative Engineering and Management Research

A Peer Reviewed Open Access International Journal

www.ijiemr.org

COPY RIGHT

2017 IJIEMR. Personal use of this material is permitted. Permission from IJIEMR must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. No Reprint should be done to this paper, all copy right is authenticated to Paper Authors

IJIEMR Transactions, online available on 30 May 2017.

Link : <http://www.ijiemr.org/downloads.php?vol=Volume-6&issue=ISSUE-3>

Title: Design of Hybrid Encoder And Decoder Using Inversion

Volume 06, Issue 03, Pages: 467 – 473.

Paper Authors

SUPRIYA .M, VASANTHA ROY.A, DURGA PRASAD. B.

Department of ECE, A.K.R.G College of Engineering Technology, Nallajerla, Andhra Pradesh, India.



USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per **UGC Guidelines** We Are Providing A Electronic Bar Code

DESIGN OF HYBRID ENCODER AND DECODER USING INVERSION

SUPRIYA .M¹, VASANTHA ROY. A², DURGA PRASAD. B³.

¹PG Scholar Department of ECE, A.K.R.G College of Engineering Technology.

²Assistant Professor Department of ECE, A.K.R.G College of Engineering Technology,

³Associate Professor & Head of Department ECE, A.K.R.G College of Engineering Technology,

ABSTRACT -In this paper, we perform a high-performance H-encoder/decoder with adaptive logic architecture for the VLSI implementation. By executing the coupling switching activity (C.S.A) obtained by different data encoding techniques (D.E.T). The proposed system called as Advanced encoder and decoder (A.E.D) is introduced. The proposed encoding is inversion operation which is restricted to some encoding schemes. As the adaptive effective encoding technique gives better delay. The Xilinx 14.7 software is used for execution with Family SPARTAN 3E. The device and package used are XC3S100E and VQ100. Preferred VHDL language to execute H-encoder/decoder with adaptive logic.

Key terms: A.E.D, C.S.A, D.E.T, Encoder, Decoder

INTRODUCTION

Encoder is a digital circuit that performs the inverse operation of decoder it has 2^n input lines and n output lines and it generates the binary code corresponding to the input values. Encoders are of two types they are incremental encoder (rotary/shaft encoder) and absolute encoder. Shafting encoder is an electro mechanical device which helps to convert the angular position or motion of a shaft or axle to an analog code or digital code.

It is a pulse generator that provides a square wave signals and a zero index[7]. To overcome this problem we gone through absolute encoder it has been developed to compensate for the performance and limitations of shaft It is a pulse generator that provides a square wave signals and a zero index. To overcome this problem we gone through absolute encoder it has been developed to compensate for the performance and limitations of shaft encoder this encoder must be reserved after a power interruption zero reset is help to obtain the mechanism angular position and sensitivity of interference[1]. This absolute encoder supplies the

shaft position as a binary code. The output code is unique for each position In this paper we focused on techniques aimed at adaptive encoding technique of power consumption by the Xilinx software here we use encoder and as well as decoder with the help of adders [4]. The proposed encoding schemes which are transparent with respect to the pulses implementation are presented and discussed at both binary/algorithmic level and the architectural level by means of simulation on synthesis and real traffic scenarios this result shows that by using the proposed encoding scheme up to 52% of power and 16% of energy can be saved without any significant degradation. However, the encoding complexity of LDPC codes is still too high, which is a major problem that needs to be solved for their implementation. There have been some studies to reduce the encoding complexity using specially formed matrices such as a lower triangular matrix or semirandom matrix [6]. The encoding process of standard LDPC codes requires transferring a parity check matrix (H) into an equivalent systematic form,

which can be accomplished by Gaussian elimination [6]. Gaussian elimination requires large memory and heavy calculation. The encoding process with a semirandom technique is much simpler than that using other matrices because it doesn't require Gaussian elimination [6]. Consequently, a Linear time encoding is possible with very little memory. The hardware implementation of LDPC decoders is another problem to be considered when we use the fully parallel decoding algorithm of LDPC codes [8]. Although the fully parallel decoders can achieve a very high decoding speed, it is too complex to implement practically [6]. One of the best solutions for the decoder architecture is to directly instantiate the belief propagation (BP) algorithm in hardware [4]. In fully parallel decoding structures, all check nodes and variable nodes have their own processors and exchange messages between each check node and variable node at the same time.

In order to lower the hardware complexity, the number of check node and variable node processors needs to be reduced. In partly parallel decoding structures, part of variable nodes and check nodes perform the message passing process in time-division multiplexing mode. Therefore, there is trade-off between decoding throughput and hardware complexity in partly parallel structures. Although the hardware complexity of LDPC decoders is reduced using the partly parallel structures, these structures have a potential problem of encoding complexity because their parity check matrices may not be suitable for an efficient encoding process.

II. EXISTING SYSTEM

Wireless mobile communication now demands large data bandwidth to accommodate various multimedia services. The third generation mobile communication, IMT-2000, provides 144 kbps for a fast moving terminal, 384 kbps for walking conditions, and 2 Mbps for stationary conditions [1]. The current bandwidth of IMT-2000 will not be enough to satisfy the various future demands for real-time and high quality service,

compared to wired communication services for fast moving stations. The specification of fourth generation (4G) mobile communication is being developed to overcome the limitation. 4G mobile communication is supposed to provide 100 Mbps for a fast moving station and 155 Mbps to 1 Gbps for slow moving and stationary conditions. Such a system requires a very high speed wireless transmission technique.

A wireless channel environment is more subject to noise than a wired channel because the signals are open to external disturbances such as path loss, shadowing, and fading. Therefore, channel coding is inevitable for wireless communication. Channel coding has been an important issue in communication systems. It has the ability to detect and correct errors caused by noise on a channel. The bit-error-rate (BER) can be reduced without increasing the signal power since the transmitted data carry redundancies that are used to detect and correct errors. This coding skill is useful in transmission on finite power channels such as general switched telephone networks [2]-[5]. To exist encoding technique I use encoder and decoder in scheme I focus on reducing the number of TYPE 1 and TYPE 2 transistors [1].

The scheme 1 compares the present data with previous one (flit) to decide whether the given input is odd inversion or no inversion of the present data can lead to power reduction if the previous bit is odd inverter before being transmitted, the dynamic power on the links are the self-transition activity of Types I, II, III and IV, respectively. For each transition the relationship between the coupling transition activities of the flit when transmitted as its bits are odd inverted. Here if the flit is odd inverted, Types II, III and IV transitions convert to the type I transitions. In case of transition types the type I transitions, the inversion of one of the Type II, III or Type IV transition respectively[1] SCHEME 2: Here I make use of both odd and full inversion the full inversion operation converts Type II to Type IV transition.

It compares the current data with the flit to decide whether the odd, full, or no inversion of the

current data can give rise to the power reduction. The power is dissipated when the flit is transmitted with no inversion, odd inversion and full inversion, respectively. The odd inversion lead to power reduction and the full inversion satisfied inequality. The operating principles of this encoder are similar to the previous encoder. Here again the previously encoded body flit is indicated with *inv* which defines if it was odd or full inverted (*inv*=1) or left as it was (*inv*=0).

In this paper, we focus on techniques aimed at reducing the power dissipated by the network links. In fact, the power dissipated by the network links is as relevant as that dissipated by routers and network interfaces (NIs) and their contribution is expected to increase as technology scales [5]. In particular, we present a set of data encoding schemes operating at flit level and on an end-to-end basis, which allows us to minimize both the switching activity and the coupling switching activity on links of the routing paths traversed by the packets. The proposed encoding schemes, which are transparent with respect to the router implementation, are presented and discussed at both the algorithmic level and the architectural level, and assessed by means of simulation on synthetic and real traffic scenarios.

The analysis takes into account several aspects and metrics of the design, including silicon area, power dissipation, and energy consumption. The results show that by using the proposed encoding schemes up to 51% of power and up to 14% of energy can be saved without any significant degradation in performance and with 15% area overhead in the NI.

Coding Principles Of FM0 Code And manchester Code

In the following discussion, the clock signal and the input data are abbreviated as CLK, and X, respectively. With the above parameters, the coding principles of FM0 and Manchester codes are discussed as follows.

FM0 encoding:

FM0 is also known as Biphase space encoding. A transition is present on every bit and an additional transition may occur in the middle of the bit. Here the data rate is twice. Sufficient clock information can be recovered from the data stream so that a separate clock is not needed. Therefore, for transmission, the number of wires is minimized. Logic 0 represents the transition in the center of the bit. Logic 1 represents there is no transition from the center of bit. This encoding data contains sufficient information to recover a clock from the data. It has to reach the DC balance and enhance signal reliability. It is used to reduce noise and transmission power. The block diagram has an XOR gate, DFF, inverter, and MUX. For example, the XOR gate has one input as feedback that is 0, and another input as 1. This XOR output is given to DFF1, and it also has a CLK signal with an output of 1. Another DFF2 has an input as 1 and CLK. The output is 1. If the selection line is 0, it produces the output as DFF1 as FM0 output. Otherwise, the selection line is 0 and produces an output as DFF2 or FM0 output. for each input, the FM0 code consists of two parts: one for former-half cycle of CLK, *A*, and the other one for later-half cycle of CLK, *B*. The coding principle of FM0 is listed as the following three rules.

- 1) If *X* is the logic-0, the FM0 code must exhibit a transition between *A* and *B*.
- 2) If *X* is the logic-1, no transition is allocated among each FM0 code no matter what the *X* is.
- 3) The transition is allocated among each FM0 code no matter what the *X* is.

A FM0 coding example is shown in Fig. 3. At cycle 1, the *X* is logic-0; therefore, a transition occurs on its FM0 code, according to rule

1. For simplicity, this transition is initially set from logic-0 to -1. According to rule 3, a transition is allocated among each FM0 code, and thereby the

logic-1 is changed to logic-0 in the beginning of cycle

2. Then, according to rule 2, this logic-level is hold without any transition in entire cycle 2 for the X of logic-1. Thus, the FM0 code of each cycle can be

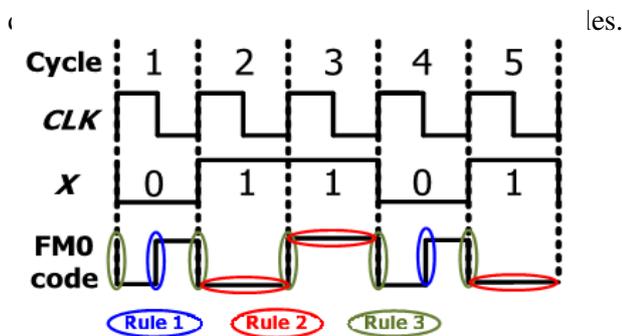


Fig1.FM0 logic state

To make an analysis on hardware utilization of FM0 and Manchester encoders, the hardware architectures of both are conducted first. As mentioned earlier, the hardware architecture of Manchester encoding is as simple as a XOR operation. However, the conduction of hardware architecture for FM0 is not as simple as that of Manchester. How to construct the hardware architecture of FM0 encoding should start with the FSM of FM0 first. As shown in Fig. 5(a), the FSM of FM0 code is classified into four states. A state code is individually assigned to each state, and each state code consists of A and B, as shown in Fig. 2. According to the coding principle of FM0, the FSM of FM0 is shown in Fig. 5(b). Suppose the initial state is S1, and its state code is 11 for A and B, respectively.

Manchester Encoding

In telecommunication and data storage, **Manchester coding** (also known as **phase encoding**, or **PE**) is a line code in which the encoding of each data bit has at least one transition and occupies the same time. It therefore has no DC component, and is self-clocking, which means that it may be inductively or capacitive coupled, and that a clock signal can be recovered from the encoded data. As a result, electrical connections using a Manchester code are easily galvanically

isolated using a network isolator a simple one-to-one isolation transformer.

Manchester coding is one of the most common data coding methods used today. Similar to Bi Phase, Manchester coding provides a means of adding the data rate clock to the message to be used on the receiving end. Also Manchester provides the added benefit of always yielding an average DC level of 50%. This has positive implications in the demodulator's circuit design as well as managing transmitted RF spectrum after modulation. This means that in modulation types where the power output is a function of the message such as AM, the average power is constant and independent of the data stream being encoded. Manchester coding states that there will always be a transition of the message signal at the mid-point of the data bit frame. What occurs at the bit edges depends on the state of the previous bit frame and does not always produce a transition. A logical "1" is defined as a mid-point transition from low to high and a "0" is a mid-point transition from high to low.

Encoding is the process of adding the correct transitions to the message signal in relation to the data that is to be sent over the communication system. The first step is to establish the data rate that is going to be

used. Once this is fixed, then the mid-bit time can be determine das $\frac{1}{2}$ of the data rate period. In our example we are going to use a data rate of 4 kHz. This provides a bit period of $1/f = 1/4000 = 0.00025s$ or 250 μs . Dividing by two gives us the mid-bittime (which we will label "T") of 125 μs . Now let's look at how we use this to encode a databyte of 0xC5 (11000101b). The easiest method to do this is to use a timer set to expire orinterrupt at the T interval. We also need to set up a method to track which $\frac{1}{2}$ bit period we are currently sending. Once we do this, we can easily encode the data and output the message signal.

1. Begin with the output signal high.
2. Check if all bits have been sent, If yes, then go to step
3. Check the next logical bit to be coded
4. If the bit equals "1", then call ManchesterOne(T)
5. Else call ManchesterZero(T)
6. Return to step 2
7. Set output signal high and return

$$X \oplus \text{CLK}$$

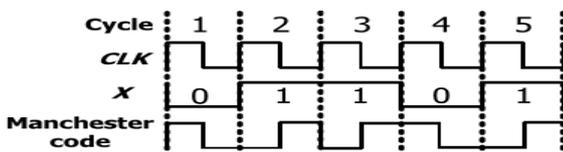


Fig 2. manchester logics states

III. PROPOSED SYSTEM

H-Encoder It is better codes of error controlling performance. H-Encoder outputs are not only associated with the encode elements at present, but also affected by several ones before. Data 1 and data 2 are used for describing codes, where data 1 are the input encode elements, data out is the output encode elements and data 2 is the shift register number of encoder Data 1 and Data 2 are the inputs of the Encoder. Their architecture design with chip registers perform their operations and gives output of the encoder is Data out as shown in fig 1.

H-Decoder Two parallel binary bits are inputted into the H-decoder with every clock pulse, and then it begins to work when the input enabling signal is valid. Each group consists of two because each current state can be reached by decoder path. Here Data out is the output of the encoder similarly Data out and Data 2' are the inputs of Decoder. These inputs will perform as per their chip

architecture design and the output of the decoder is Data as shown in fig-1.

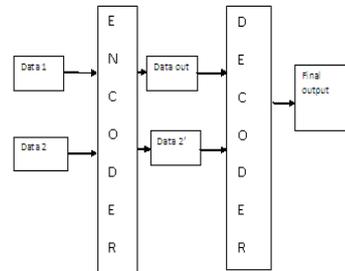
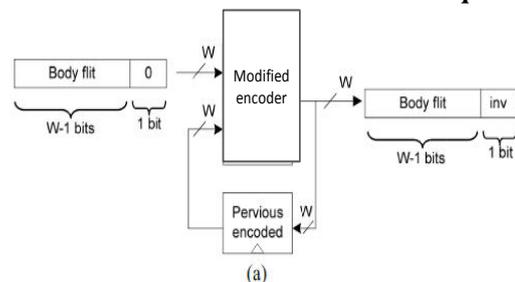


Fig.3.H-Encoder/Decoder with Adaptive Logic

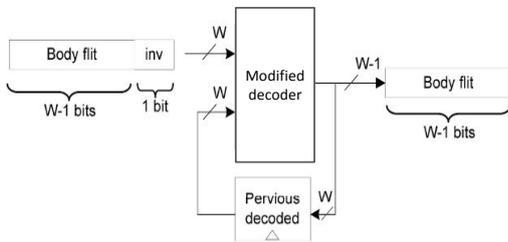
Figure 1 indicates the block diagram of H-encoder/decoder with adaptive logic. The H-encoder/decoder with adaptive logic consists of four shift registers and two exclusive-or gates. Every shift register is equivalent to a flip flop. These four flip flops are connected in series to complete shifting and updating operation under the action of the clock pulse. The exclusive-or gates are used for inner operation of coding data. With every clock pulse the encoder outputs two bits according to the generator polynomials whenever one binary bit is inputted. The output is not only relevant with the current input binary bit, but also influenced by

Proposed system

Encoder Architecture with H Technique



Decoder Architecture with H Technique



Comparison of Parameters

Parameter s	Total memory used	Total delay	Logic delay
Existing system ^[1] [Manchester]	518600k b	15.753n s	240%
Proposed system [H-encoder]	210768k b	5.058ns	7.5%

IV .RESULTS

TECHNICAL SCHEMATIC:

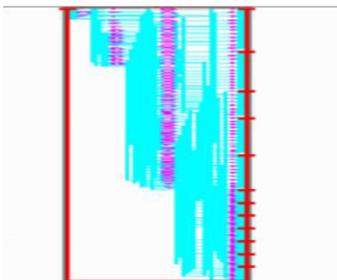


FIG.5.Technical Schematic

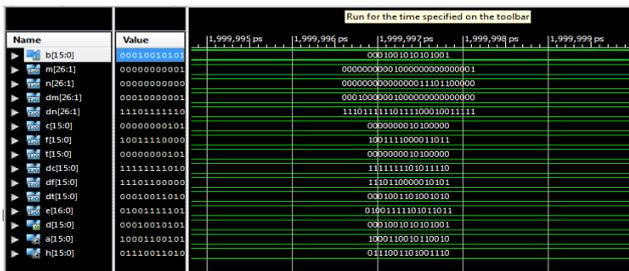


FIG.3.Output Waveforms

The H-Encoder/Decoder with Adaptive logic decides the proposed algorithm to work with adaptive logic. The registers are work with the adaptive techniques, by using this technique, the proposed algorithm increases the speed i.e. decrease the delay.

V CONCLUSION

A.E.D is based on the proposition that Genetic algorithm finds good solutions to a problem. The proposed encoding supports one point crossover as in binary encoding and A.E.D crossover as in encoding. The proposed encoding is inversion operation which is restricted to some encoding schemes. A.E.D is very efficient in speed. As the adaptive effective encoding technique gives better delay.

VI REFERENCES

- [1] Yu-hsuan Lee, member, IEEE, and cheng-wei pan “fully reused VLSI architecture of FMO/Manchester encoding using SOLS technique for DSRC applications” IEEE transactions on very large scale integration(VLSI)systems, vol.,23,no.1,January 2015
- [2]. NimaJafarzadeh, Maurizio Palesi and Ali Afzali-Kusha “Data Encoding Techniques for Reducing Energy Consumption in Networkon-Chip” IEEE transactions on very large scale integration (vlsi) systems, vol. 22, no. 3, march 2014

- [3]. M. Palesi, R. Tornero, J. M. Orduña, V. Catania, and D. Panno, "Designing robust routing algorithms and mapping cores in networksonchip: A multi-objective evolutionary-based approach," *J. Univ. Comput. Sci.*, vol. 18, no. 7, pp. 937–969, 2006.
- [4]. M. S. Rahaman and M. H. Chowdhury, "Crosstalk avoidance and errorcorrection coding for coupled RLC interconnects," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2009, pp. 141–144. [1] F. Ahmed-Zaid, F. Bai, S.
- [5]. W. Wolf, A. A. Jerraya, and G. Martin, "Multiprocessor system-onchipMPSoC technology," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 10, pp. 1701–174, Oct. 2008.
- [6]. L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002. [6]. S. E. Lee and N. Bagherzadeh, "A variable frequency link for a poweraware network-on-chip (NoC)," *Integr. VLSI J.*, vol. 42, no. 4, pp. 479–485, Sep. 2009.
- [7]. S. R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, W. James, D. Finan, A. P. Singh, T. Jacob, S. Jain, V. Erraguntla, C.Roberts, Y. V. Hoskote, N. Y. Borkar, and S. Y. Borkar, "An 80-tile Sub-100-W TeraFLOPS processor in 65-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 29–41, Jan. 2008.
- [8]. S. Murali, C. Seiculescu, L. Benini, and G. De Micheli, "Synthesis of networks on chips for 3D systems on chips," in *Proc. Asia South Pacific Design Autom. Conf.*, Jan. 2009, pp. 242–247.